IPW

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/664,418	09/17/2003	Bruce D. Holenstein	9203-27U1	3213

570 7590 12/13/2005

AKIN GUMP STRAUSS HAUER & FELD L.L.P.
ONE COMMERCE SQUARE
2005 MARKET STREET, SUITE 2200
PHILADELPHIA, PA 19103

EXAMINER	
HICKS, MICHAEL J	
ART UNIT	**PAPER NUMBER**
2165	

DATE MAILED: 12/13/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)
	10/664,418	HOLENSTEIN ET AL.
	Examiner	**Art Unit**
	Michael J. Hicks	2165

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 September 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *17 September 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *2/12/2004*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-9 are pending in the instant application.

Double Patenting

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., In re Berg, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); In re Goodman, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); In re Longi, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); In re Van Ornum, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); In re Vogel, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and In re Thorington, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 1-36 of U.S. Patent 6,662,196 contain every element of claims 1, 2, 4, 6, and 8 of the instant application and as such anticipate Claims 1, 2, 4, 6, and 8 of the instant application.

Claims 1-39 of U.S. Application Number 10/112129 contain every element of claims 1, 2, 4, 6, and 8 of the instant application and as such anticipate Claims 1, 2, 4, 6, and 8 of the instant application.

"A later patent claim is not patentably distinct from an earlier patent claim if the later claim is obvious over, or **anticipated by**, the earlier claim. In re Longi, 759 F.2d at 896, 225 USPQ at 651 (affirming a holding of obviousness-type double patenting because the claims at issue were obvious over claims in four prior art patents); In re Berg, 140 F.3d at 1437, 46 USPQ2d at 1233 (Fed. Cir. 1998) (affirming a holding of obviousness-type double patenting where a patent application claim to a genus is anticipated by a patent claim to a species within that genus). " ELI LILLY AND COMPANY v BARR LABORATORIES, INC., United States Court of Appeals for the Federal Circuit, ON PETITION FOR REHEARING EN BANC (DECIDED: May 30, 2001).

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 3 and 5 rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Regarding Claims 3 and 5, the phrase "multi-threaded manner" renders the claim indefinite because it is unclear whether the limitations of the claim are restricted solely to performing the operations using multi-threading, or if the claim encompasses other limitations as well. Specifically, the word "manner" creates the indefiniteness. See MPEP § 2173.05(d).

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7. Claims 1, 2, and 4 rejected under 35 U.S.C. 102(e) as being anticipated by Bortvedt et al. (US Patent Number 5,799,305 and referred to hereinafter as Bortvedt).

As per Claim 1, Bortvedt discloses a method of replicating data associated with a plurality of transactions (i.e. *"The present invention is directed to a method of committing a distributed transaction in a distributed database system."* The preceding text excerpt clearly indicates that the data associated with transactions is distributed/replicated in many databases of a distributed database system.) (Column 4, Lines 1-13) in a replication system including a plurality of nodes connected via communication media in a topology (i.e. *"The present invention is directed to a method of committing a distributed transaction in a distributed database system."* The preceding text excerpt clearly indicates that the transactions are executed in a distributed database system/a plurality of nodes connected via communication media in a topology.) (Column 4, Lines 1-13), each node including a database, at least some of the nodes being able to independently receive and post transactions (i.e. *"The present invention is directed to a method of committing a distributed transaction in a distributed database system."* The preceding text excerpt clearly indicates that the transactions are executed in a distributed database system further indicating that each node includes a

database and that each database is able to independently receive and post transactions (e.g. distribute queries).) (Column 4, Lines 1-13), the method comprising: (a) initiating and performing transactions to be executed in a database at an originating node (i.e. *"A method for committing a distributed transaction in a distributed database."* The preceding text excerpt clearly indicates that an originating node may commit a transaction, which further indicates that the transaction was initiated and executed.) (Abstract); and (b) replicating the transactions to at least one or more other nodes by: (i) pausing each transaction being executed in the database at the originating node prior to a commit operation for the transaction (i.e. *"...if an external transaction enters the REQUEST COMMIT state...the IC places the external transaction into a PUASED state."* The preceding text excerpt clearly indicates that each transaction is placed into a paused state/is paused after a request to commit/prior to committing.) (Column 28, Lines 62-65), (ii) assigning a ready to commit token to the transaction (i.e. *"...to execute a credit operation on checking accounts table, owner transmits request message to coserver..."* The preceding text excerpt clearly indicates that a request message/ready to commit token is assigned to the operation/transaction.) (Column 12, Lines 5-7), (iii) sending the ready to commit token to the one or more other nodes (i.e. *"...to execute a credit operation on checking accounts table, owner transmits request message to coserver..."* The preceding text excerpt clearly indicates that the request message/ready to commit token is sent to each coserver/one or more other nodes.) (Column 12, Lines 5-7), (iv) determining at the one or more other nodes whether the respective databases are prepared for a commit operation for the transaction corresponding to the ready to commit token, and, if so, sending back the ready to commit token to the originating node (i.e. *"Once coserver has completed execution of the credit operation, at time A, it enters log record in a transaction log...After coserver enters log record into transaction log, it sends a completion message to owner..."* The preceding text excerpt clearly indicates that the coserver/one or more other nodes

determine whether they are ready/prepared to commit to the transaction indicated by the request message/ready to commit token and send a completion message/ready to commit token back to the owner/originating node only after the operation/transaction is complete and prepared to commit.) (Column 12, Lines 11-20), and (v) executing a commit operation at the database of the originating node only upon receipt from at least one of the other nodes of the ready to commit tokens originally sent from the originating node (i.e. *"...the owner examines each completion message and...the transaction is marked as ready to commit."* The preceding text excerpt clearly indicates that when the coservers/one or more of the other nodes submits a completion message corresponding to the request message sent by the owner, the commit operation is executed. Note that there are several intermediate steps before the transaction is committed, but the commit process does not begin until the coservers/one or more other nodes submits a completion message.) (Column 12, Lines 36-30, 62-63), wherein step (b) is performed asynchronously with respect to other transactions that are initiated in step (a) (i.e. *"A list of transactions that have been committed is included in the next interval message."* The preceding text excerpt clearly indicates because multiple transactions may commit in one commit interval and because net every transaction takes the same amount of time to commit, multiple transaction may be running asynchronously, e.g. transactions may be initiated while others are committing.) (Column 29, Lines 40-41).

As per Claim 2, Bortvedt further discloses the commit operation in step (b)(v) is performed only upon receipt from each of the one or more the other nodes of the ready to commit tokens originally sent from the originating node (i.e. *"The IC can commit a transaction once it determines that all of the participants in the transaction have sent a closure message..."* The preceding text excerpt clearly indicates the transaction is committed only after all/each of the participants/one or more other nodes has responded with a closure message/ready to commit token.) (Column 29, Lines 34-38).

As per Claim 4, Bortvedt discloses a method of replicating data associated with a plurality of (i.e. *"The present invention is directed to a method of committing a distributed transaction in a distributed database system."* The preceding text excerpt clearly indicates that the data associated with transactions is distributed/replicated in many databases of a distributed database system.) (Column 4, Lines 1-13) in a replication system including a plurality of nodes connected via communication media in a topology (i.e. *"The present invention is directed to a method of committing a distributed transaction in a distributed database system."* The preceding text excerpt clearly indicates that the transactions are executed in a distributed database system/a plurality of nodes connected via communication media in a topology.) (Column 4, Lines 1-13), each node including (i) a database i.e. *"The present invention is directed to a method of committing a distributed transaction in a distributed database system."* The preceding text excerpt clearly indicates that the transactions are executed in a distributed database system, therefore each node includes a database.) (Column 4, Lines 1-13), (ii) a replication engine which performs data replication functions (i.e. *"...to execute a credit operation on checking accounts table, owner transmits request message to coserver..."* The preceding text excerpt clearly indicates that the there is a mechanism at the nodes/coservers which is able to create and send message/tokens (e.g. perform data replication functions) which can be considered a data replication engine.) (Column 12, Lines 5-7), and (iii) an application which executes transactions and posts the transactions to the database, the application being independent of the replication engine, each transaction being one or more transaction steps or transaction operations (i.e. *"The interval coordinator is a program which determines when a transaction is committed...These interval coordinator and interval participant may be parts of, subroutines of, or separate programs callable by coservers."* The preceding text excerpt clearly indicates that an application exists on the owner node, called the interval coordinator, which is not tied to the replication engine, that executes and post transactions to the database, where the transaction must

consist of at least one transaction step/operation.) (Column 7, Lines 56-62), the method comprising: (a) an application at a first node that initiating and performs transactions to be executed in a database at an originating node, the application pausing each transaction being executed in a source database at the first node prior to a commit operation for the transaction (i.e. *"...if an external transaction enters the REQUEST COMMIT state...the IC places the external transaction into a PUASED state."* The preceding text excerpt clearly indicates that each transaction from the originating node is placed into a paused state/is paused after a request to commit/prior to committing. Note that if the transaction is taking place, the an application at the node must have initiated and performed the transaction.) (Column 28, Lines 62-65); (b) a replication engine at the first node assigning a ready to commit token to the transaction in coordination with the application (i.e. *"...to execute a credit operation on checking accounts table, owner transmits request message to coserver..."* The preceding text excerpt clearly indicates that a request message/ready to commit token is assigned to the operation/transaction by the replication engine at the owner/first node.) (Column 12, Lines 5-7); (c) the replication engine at the first node sending the ready to commit token to the second node (i.e. *"...to execute a credit operation on checking accounts table, owner transmits request message to coserver..."* The preceding text excerpt clearly indicates that the request message/ready to commit token is sent to each coserver/second node.) (Column 12, Lines 5-7); (d) a replication engine at a second node determining whether a target database at the second node is prepared for a commit operation for the transaction corresponding to the ready to commit token, and, if so, sending back the ready to commit token to the first node (i.e. *"Once coserver has completed execution of the credit operation, at time A, it enters log record in a transaction log...After coserver enters log record into transaction log, it sends a completion message to owner..."* The preceding text excerpt clearly indicates that the replication engine at the coserver/second node determines whether they are ready/prepared to

commit to the transaction indicated by the request message/ready to commit token and send a completion message/ready to commit token back to the owner/originating node only after the operation/transaction is complete and prepared to commit.) (Column 12, Lines 11-20); and (e) the application at the first node executing a commit operation at the source database in coordination with the replication engine only upon receipt from the second node of the ready to commit token originally sent from the first node (i.e. *"...the owner examines each completion message and...the transaction is marked as ready to commit...when the IC receives the closure message, it determines that the account transfer transaction is eligible for commit."* The preceding text excerpt clearly indicates that when the coservers/second node submits a completion message corresponding to the request message sent by the owner, the commit operation is executed by the application. Note that there are several intermediate steps before the transaction is committed, but the commit process does not begin until the coservers/one or more other nodes submits a completion message.) (Column 12, Lines 36-30, 62-63), wherein the replication engine operates asynchronously with respect to the application until step (b) occurs (i.e. *"A list of transactions that have been committed is included in the next interval message."* The preceding text excerpt clearly indicates because multiple transactions may commit in one commit interval and because net every transaction takes the same amount of time to commit, multiple transaction may be running asynchronously, e.g. transactions may be initiated while others are committing.) (Column 29, Lines 40-41).

8. Claims 6-9 rejected under 35 U.S.C. 102(b) as being anticipated by Holliday et al. ("The performance of database replication with group multicast", Fault-Tolerant Computing, 1999. Digest of Papers. Twenty-Ninth Annual International Symposium on 15-18 June 1999 Page(s):158 – 165 and reffered to hereinafter as Holliday).

As per Claim 6, Holliday discloses a method of performing dual writes for replicating transactions among plural databases located at different nodes (i.e. *"We*

therefore consider replicating several copies of the database (or perhaps only the hot-spot pages) on multiple sites connected by a local area network...we assume that concurrency control is locally enforced by strict two phase locking at all server sites." The preceding text excerpt clearly indicates that a replicated database with database copies located at different sites/nodes exists which utilized two-phase locking/dual writes.) (Page 159, Column 2, Paragraph 4; Page 159, Column 1, Paragraphs 1-2), each transaction being one or more transaction steps or transaction operations (i.e. *"This is achieved by deferring update operations until commit time, when a single message with all updates is sent to all other sites."* The preceding text excerpt clearly indicates that each transaction is a collection of/one or more read and write operations (e.g. transaction operations.) (Page 160, Column 1, Paragraph 2), at least some of the transaction steps or transaction operations being update steps or operations (i.e. *"This is achieved by deferring update operations until commit time, when a single message with all updates is sent to all other sites."* The preceding text excerpt clearly indicates that at least some of the transaction operations are write operations (e.g. update operations).) (Page 160, Column 1, Paragraph 2), the method comprising: (a) initiating a transaction at an originating node (i.e. *"A transaction Ti, executes a read operation locally, while a write operation is deferred until Ti is ready to commit."* The preceding text excerpt clearly indicates that the collection of read and write operations/transaction is initiated at an originating node.) (Page 160, Column 1, Paragraph 2); (b) inhibiting the dual write replication process from communicating transaction steps or operations of the transaction with one or more other nodes until an update step or operation occurs within the transaction at the originating node (i.e. *"A transaction Ti, executes a read operation locally, while a write operation is deferred until Ti is ready to commit. To terminate, Ti broadcasts its deferred writes to all sites."* The preceding text excerpt clearly indicates that the broadcast of write operations/beginning of the dual write replication process is delayed until the write/update operations occur.) (Page 160, Column 1, Paragraph 2); and (c) upon the occurrence

of the update step or operation, performing the dual write replication process with respect to the one or more other nodes and sending with the update step or operation all transaction steps or operations for the transaction that have occurred prior to the update step or operation for the transaction, or prior to the previous update step or operation if a previous update step or operation existed for the transaction (i.e. *"To terminate, Ti broadcasts it's deferred writes to all sites. On receiving the writes, the lock manager on site S grants all write locks to Ti atomically, and then the writes are executed at S. After all the writes of Ti are executed locally, Ti broadcasts its commit operation to all sites. Ti terminates after the delivery and execution of its commit locally."* The preceding text excerpt clearly indicates that the two phase lock operation/dual write replication process is executed with respect to the other sites/one or more other nodes, and that all writes/transaction operations which have occurred prior to the update operation are sent during the execution.) (Page 160, Column 1, Paragraph 2).

As per Claims 7 and 9, Holliday further discloses (d) determining if the originating node needs to receive a data record from the one or more other nodes during the dual write replication process (i.e. *"To terminate, Ti broadcasts its deferred writes to all sites."* The preceding text excerpt clearly indicates that because the update operations are write operations, the one or more other sites/nodes will always need to receive a data record.) (Page 160, Column 1, Paragraph 2); and (e) sending the data record to the originating node only if it is determined that the originating node needs the data record (i.e. *"To terminate, Ti broadcasts its deferred writes to all sites...After all writes of Ti are executed locally, Ti broadcasts it's commit operation to all sites."* The preceding text excerpt clearly indicates that if the one or more other sites/nodes are to commit a write operation, the record that was needed for the write operation must have been sent in the broadcast from the originating site/node.) (Page 160, Column 1, Paragraph 2).

As per Claim 8, Holliday discloses a method of performing dual writes for replicating transactions among plural databases located at different nodes (i.e. *"We therefore consider replicating several copies of the database (or perhaps only the hot-spot pages) on multiple sites connected by a local area network...we assume that concurrency control is locally enforced by strict two phase locking at all server sites."* The preceding text excerpt clearly indicates that a replicated database with database copies located at different sites/nodes exists which utilized two-phase locking/dual writes.) (Page 159, Column 2, Paragraph 4; Page 159, Column 1, Paragraphs 1-2), each transaction being one or more transaction steps or transaction operations (i.e. *"This is achieved by deferring update operations until commit time, when a single message with all updates is sent to all other sites."* The preceding text excerpt clearly indicates that each transaction is a collection of/one or more read and write operations (e.g. transaction operations.) (Page 160, Column 1, Paragraph 2), at least some of the transaction steps or transaction operations being update steps or operations which are performed only after database locking occurs (i.e. *"This is achieved by deferring update operations until commit time, when a single message with all updates is sent to all other sites. On receiving the writes, the lock manager on site S grants all write locks to Ti atomically, and then the writes are executed at S."* The preceding text excerpt clearly indicates that at least some of the transaction operations are write operations (e.g. update operations), and that the update operations execute/occur only after database locking occurs.) (Page 160, Column 1, Paragraph 2), the method comprising: (a) initiating a transaction at an originating node (i.e. *"A transaction Ti, executes a read operation locally, while a write operation is deferred until Ti is ready to commit."* The preceding text excerpt clearly indicates that the collection of read and write operations/transaction is initiated at an originating node.) (Page 160, Column 1, Paragraph 2); (b) the dual write replication process causing database locking to occur at one or more other nodes only upon the occurrence of an update step or operation in the transaction at the originating node. (i.e. *"A transaction Ti, executes a read operation locally, while a write operation is deferred until Ti is ready to commit. To*

terminate, Ti broadcasts its deferred writes to all sites. On receiving the writes, the lock manager on site S grants all write locks to Ti atomically, and then the writes are executed at S." The preceding text excerpt clearly indicates that the database locking in the two phase lock process/dual write replication process is delayed until the write/update operations occur at the originating node.) (Page 160, Column 1, Paragraph 2).

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 3 and 5 rejected under 35 U.S.C. 103(a) as being unpatentable over Bortvedt in view of Arevalo et al. ("Deterministic Scheduling for Transactional Multithreaded Replicas", 19th IEEE Symposium on Reliable Distributed Systems (SRDS'00) p. 164, 2000 and referred to hereinafter as Arevalo).

As per Claim 3, Bortvedt discloses a method of replicating data associated with a plurality of transactions (i.e. *"The present invention is directed to a method of committing a distributed transaction in a distributed database system."* The preceding text excerpt clearly indicates that the data associated with transactions is distributed/replicated in many databases of a distributed database system.) (Column 4, Lines 1-13) in a replication system including a plurality of nodes connected via communication media in a topology (i.e. *"The present invention is directed to a method of committing a distributed transaction in a distributed database system."* The preceding text excerpt clearly indicates that the transactions are executed in a distributed database system/a plurality of nodes connected via communication media in a topology.) (Column 4, Lines 1-13), each node

including a database, at least some of the nodes being able to independently receive and post transactions (i.e. *"The present invention is directed to a method of committing a distributed transaction in a distributed database system."* The preceding text excerpt clearly indicates that the transactions are executed in a distributed database system further indicating that each node includes a database and that each database is able to independently receive and post transactions (e.g. distribute queries).) (Column 4, Lines 1-13), the method comprising: (a) initiating and performing transactions to be executed in a database at an originating node (i.e. *"A method for committing a distributed transaction in a distributed database."* The preceding text excerpt clearly indicates that an originating node may commit a transaction, which further indicates that the transaction was initiated and executed.) (Abstract); and (b) replicating the transactions to at least one or more other nodes by: (i) pausing each transaction being executed in the database at the originating node prior to a commit operation for the transaction (i.e. *"...if an external transaction enters the REQUEST COMMIT state...the IC places the external transaction into a PUASED state."* The preceding text excerpt clearly indicates that each transaction is placed into a paused state/is paused after a request to commit/prior to committing.) (Column 28, Lines 62-65), (ii) assigning a ready to commit token to the transaction (i.e. *"...to execute a credit operation on checking accounts table, owner transmits request message to coserver..."* The preceding text excerpt clearly indicates that a request message/ready to commit token is assigned to the operation/transaction.) (Column 12, Lines 5-7), (iii) sending the ready to commit token to the one or more other nodes (i.e. *"...to execute a credit operation on checking accounts table, owner transmits request message to coserver..."* The preceding text excerpt clearly indicates that the request message/ready to commit token is sent to each coserver/one or more other nodes.) (Column 12, Lines 5-7), (iv) determining at the one or more other nodes whether the respective databases are prepared for a commit operation for the transaction corresponding to the ready to

commit token, and, if so, sending back the ready to commit token to the originating node (i.e. *"Once coserver has completed execution of the credit operation, at time A, it enters log record in a transaction log...After coserver enters log record into transaction log, it sends a completion message to owner..."* The preceding text excerpt clearly indicates that the coserver/one or more other nodes determine whether they are ready/prepared to commit to the transaction indicated by the request message/ready to commit token and send a completion message/ready to commit token back to the owner/originating node only after the operation/transaction is complete and prepared to commit.) (Column 12, Lines 11-20), and (v) executing a commit operation at the database of the originating node only upon receipt from at least one of the other nodes of the ready to commit tokens originally sent from the originating node (i.e. *"...the owner examines each completion message and...the transaction is marked as ready to commit."* The preceding text excerpt clearly indicates that when the coservers/one or more of the other nodes submits a completion message corresponding to the request message sent by the owner, the commit operation is executed. Note that there are several intermediate steps before the transaction is committed, but the commit process does not begin until the coservers/one or more other nodes submits a completion message.) (Column 12, Lines 36-30, 62-63), wherein step (b) is performed asynchronously with respect to other transactions that are initiated in step (a) (i.e. *"A list of transactions that have been committed is included in the next interval message."* The preceding text excerpt clearly indicates because multiple transactions may commit in one commit interval and because net every transaction takes the same amount of time to commit, multiple transaction may be running asynchronously, e.g. transactions may be initiated while others are committing.) (Column 29, Lines 40-41).

Bortvedt fails to disclose the transactions are initiated and performed in a multi-threaded manner.

Arevalo discloses the transactions are initiated and performed in a multi-threaded manner (i.e. *"In this paper, we present a deterministic scheduling algorithm for multithreaded replicas in*

a transactional framework." The preceding text excerpt clearly indicates that replicas are made through transactions in a multi-threaded manner.) (Page 164, Abstract).

It would have been obvious to one skilled in the art at the time of Applicants invention to combine the teachings of Bortvedt with the teachings of Arevalo to include the transactions are initiated and performed in a multi-threaded manner with the motivation to be able to execute several transactions concurrently.

As per Claim 5, Bortvedt discloses a method of replicating data associated with a plurality of (i.e. *"The present invention is directed to a method of committing a distributed transaction in a distributed database system."* The preceding text excerpt clearly indicates that the data associated with transactions is distributed/replicated in many databases of a distributed database system.) (Column 4, Lines 1-13) in a replication system including a plurality of nodes connected via communication media in a topology (i.e. *"The present invention is directed to a method of committing a distributed transaction in a distributed database system."* The preceding text excerpt clearly indicates that the transactions are executed in a distributed database system/a plurality of nodes connected via communication media in a topology.) (Column 4, Lines 1-13), each node including (i) a database i.e. *"The present invention is directed to a method of committing a distributed transaction in a distributed database system."* The preceding text excerpt clearly indicates that the transactions are executed in a distributed database system, therefore each node includes a database.) (Column 4, Lines 1-13), (ii) a replication engine which performs data replication functions (i.e. *"...to execute a credit operation on checking accounts table, owner transmits request message to coserver..."* The preceding text excerpt clearly indicates that the there is a mechanism at the nodes/coservers which is able to create and send message/tokens (e.g. perform data replication functions) which can be considered a data replication engine.) (Column 12, Lines 5-7), and (iii) an application which executes transactions and posts the transactions to the database, the application being

independent of the replication engine, each transaction being one or more transaction steps or transaction operations (i.e. *"The interval coordinator is a program which determines when a transaction is committed...These interval coordinator and interval participant may be parts of, subroutines of, or separate programs callable by coservers."* The preceding text excerpt clearly indicates that an application exists on the owner node, called the interval coordinator, which is not tied to the replication engine, that executes and post transactions to the database, where the transaction must consist of at least one transaction step/operation.) (Column 7, Lines 56-62), the method comprising: (a) an application at a first node that initiating and performs transactions to be executed in a database at an originating node, the application pausing each transaction being executed in a source database at the first node prior to a commit operation for the transaction (i.e. *"...if an external transaction enters the REQUEST COMMIT state...the IC places the external transaction into a PUASED state."* The preceding text excerpt clearly indicates that each transaction from the originating node is placed into a paused state/is paused after a request to commit/prior to committing. Note that if the transaction is taking place, the an application at the node must have initiated and performed the transaction.) (Column 28, Lines 62-65); (b) a replication engine at the first node assigning a ready to commit token to the transaction in coordination with the application (i.e. *"...to execute a credit operation on checking accounts table, owner transmits request message to coserver..."* The preceding text excerpt clearly indicates that a request message/ready to commit token is assigned to the operation/transaction by the replication engine at the owner/first node.) (Column 12, Lines 5-7); (c) the replication engine at the first node sending the ready to commit token to the second node (i.e. *"...to execute a credit operation on checking accounts table, owner transmits request message to coserver..."* The preceding text excerpt clearly indicates that the request message/ready to commit token is sent to each coserver/second node.) (Column 12, Lines 5-7); (d) a replication engine at a second node determining whether a

target database at the second node is prepared for a commit operation for the transaction corresponding to the ready to commit token, and, if so, sending back the ready to commit token to the first node (i.e. *"Once coserver has completed execution of the credit operation, at time A, it enters log record in a transaction log...After coserver enters log record into transaction log, it sends a completion message to owner..."* The preceding text excerpt clearly indicates that the replication engine at the coserver/second node determines whether they are ready/prepared to commit to the transaction indicated by the request message/ready to commit token and send a completion message/ready to commit token back to the owner/originating node only after the operation/transaction is complete and prepared to commit.) (Column 12, Lines 11-20); and (e) the application at the first node executing a commit operation at the source database in coordination with the replication engine only upon receipt from the second node of the ready to commit token originally sent from the first node (i.e. *"...the owner examines each completion message and...the transaction is marked as ready to commit...when the IC receives the closure message, it determines that the account transfer transaction is eligible for commit."* The preceding text excerpt clearly indicates that when the coservers/second node submits a completion message corresponding to the request message sent by the owner, the commit operation is executed by the application. Note that there are several intermediate steps before the transaction is committed, but the commit process does not begin until the coservers/one or more other nodes submits a completion message.) (Column 12, Lines 36-30, 62-63), wherein the replication engine operates asynchronously with respect to the application until step (b) occurs (i.e. *"A list of transactions that have been committed is included in the next interval message."* The preceding text excerpt clearly indicates because multiple transactions may commit in one commit interval and because net every transaction takes the same amount of time to commit, multiple transaction may be running asynchronously, e.g. transactions may be initiated while others are committing.) (Column 29, Lines 40-41).

Bortvedt fails to disclose the transactions are initiated and performed in a multi-threaded manner.

Arevalo discloses the transactions are initiated and performed in a multi-threaded manner (i.e. *"In this paper, we present a deterministic scheduling algorithm for multithreaded replicas in a transactional framework."* The preceding text excerpt clearly indicates that replicas are made through transactions in a multi-threaded manner.) (Page 164, Abstract).

It would have been obvious to one skilled in the art at the time of Applicants invention to combine the teachings of Bortvedt with the teachings of Arevalo to include the transactions are initiated and performed in a multi-threaded manner with the motivation to be able to execute several transactions concurrently.

Points of Contact

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Hicks whose telephone number is (571) 272-2670. The examiner can normally be reached on Monday - Friday 8:30a - 5:00p.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Gaffin can be reached on (571) 272-4146. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael J Hicks
Art Unit 2165
(571) 272-2670